

REMARKS

Please reconsider the application in view of the above amendments and the following remarks. Applicant thanks the Examiner for carefully considering this application.

Drawings

Applicant respectfully requests the Examiner to indicate whether the formal drawings submitted on June 25, 2003, are acceptable.

Disposition of Claims

Claims 1-17 are pending in this application. Claims 1, 13, 16, and 17 are independent. The remaining claims depend, directly or indirectly, from claim 1 or claim 13.

Claim Amendments

Independent claims 1, 13, 16, and 17 have been amended to remove instances of “read/write.” No new matter has been added by way of the amendments to claims 1, 13, 16, or 17, as support for these amendments is present in the claims.

Additionally, claim 8 has been amended by way of this reply in accordance with the Examiner’s suggestion on July 24, 2006. Specifically, claim 8 has been amended to clarify that the database schema defines how the attribute is applied within the plurality of states and the plurality of transitions using the application usage specification. No new subject matter has been added by way of these amendments, as support for these amendments is present, for example, in claim 8 and in paragraphs [0020]-[0021] of the present application.

Objection(s)

Claims 1, 13, 16, and 17 are objected to for informalities. Specifically, the Examiner asserts that the recitation “read/write” in the claims makes the claims unspecified as to whether the consistency is read only, write only, or both. By way of this reply, the claims have been amended to remove instances of “read/write.” Accordingly, Applicant respectfully asserts that claims 1, 13, 16, and 17 are specific, and withdrawal of the objection to the claims is respectfully requested.

Claims 8 is objected to by the Examiner for reciting “to be used” in the body of the claim. By way of this reply, claim 8 has been amended to recite “applied,” in accordance with the Examiner’s suggestion. Accordingly, claim 8 is recited in a definite form, and withdrawal of the objection to claim 8 is respectfully requested.

Rejection(s) under 35 U.S.C. § 102

Claims 1-17 are rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,615,362 (“Jensen”). To the extent that this rejection applies to the amended claims, this rejection is respectfully traversed.

Under 35 U.S.C. §102(b), a person shall be entitled to a patent unless “the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of the application for patent in the United States,” Furthermore:

Anticipation under 35 U.S.C. § 102 means lack of novelty, and is a question of fact. To anticipate, every element and limitation of the claimed invention must be found in a single prior art reference, arranged as in the claim.

Brown v. 3M, 265 F.3d 1349, 1351 (Fed. Cir. 2001) (emphasis added). The Federal Circuit has held repeatedly that anticipation requires disclosure of each and every element of the claimed invention in a single prior art reference. See, e.g., *Schering Corp. v. Geneva Pharms.*, 339 F.3d 1373, 1377 (Fed. Cir. 2003); *Diversitech Corp. v. Century Steps, Inc.*, 850 F.2d 675, 677 (Fed. Cir. 1988); *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1574 (Fed. Cir. 1986).

A. Claims 1-12

Claim 1 is representative of this group. Accordingly, the following arguments with respect to claim 1 are similarly applicable to claims 2-12.

Jensen does not disclose a transaction or a state

Independent claim 1 requires, in part: “an application comprising a transaction, wherein the transaction comprises at least one of a plurality of states, at least one of a plurality of transitions, and at least one artifact.”

This limitation requires that *the application* includes a transaction, where the transaction includes a state, a transition, and an artifact. Further, the state corresponds to the state of the application. Said another way, the application at any given time is in a particular state.

Turning to the rejection, the Examiner has asserted that transaction recited in the claim corresponds to an object instance (see Office Action mailed July 24, 2006, p. 3). Applicant respectfully disagrees. Specifically, Jensen defines an object instance as follows:

“An “object instance” is an embodiment (instantiation) of an object class. Instances are differentiated from one another by their

attribute values, but not their routines (capabilities). For example, Jane Smith may be a first person-object instance and John Doe may be a second person-object instance." (Jensen, col. 5, ll. 50-57)

From the above definition, the object instance is not equivalent to a transaction. Further, it would be improper for the Examiner to read the term "object instance" any broader than the specific (and unambiguous) definition provided within the cited prior art. Specifically, an object instance corresponds to a runtime instance of a class, which is composed of attributes. In contrast, the transaction, as recited in the claim, includes states and transitions, where the states are associated with the application. There is no mention that the object instance, as defined in Jensen, includes or contemplates a state or a transition, where the state is associated with an application. Again, any attempt by the Examiner to read the term clearly defined in Jensen more broadly is improper.

The Examiner has also asserted that the term "state" as recited in the claims is equivalent to the term "state" that appears in Jensen (*see* Office Action mailed July 24, 2006, p. 3). Applicant respectfully disagrees. In particular, the term "state" used in Jensen corresponds to the state of a given piece of data (*see* Jensen, col. 9, ll. 20-26). In contrast, the term "state" as recited in the claims corresponds to the state of the application. Thus, the term "state" used in Jensen is clearly not equivalent to the term "state" recited in the claims.

Moreover, the following passage of Jensen makes it clear that the term "state" is used to refer to the state of a given piece of data of an object instance, and not to a state of an application as required by the claimed invention.

In the illustrated embodiment, each object instance also contains a reference count and state. Department instance 201 has attribute 203 which contains a reference count of 2, indicating that two variables in the object-oriented application refer to department instance 201, and a state of 1, indicating that the data associated

with department instance 201 is valid, i.e., has been read since the last database transaction was committed. Employee instance 211 contains a reference count of 1 and a state of 1 in attribute 213. The reference count of 1 indicates that only one variable in the object-oriented application refers to employee instance 211. Employee instance 218 contains a reference count of 1 and a state of 0 in attribute 220. The state of 0 indicates that the data associated with employee instance 218 has been flushed, i.e., has not been read since the last database transaction was committed. The reference count and state can be omitted in some embodiments (Jensen, col. 9, lines 23-35)

Jensen does not disclose a consistency specification

Independent claim 1 additionally requires, in part, that “wherein the consistency specification specifies at least one of a read consistency and a write consistency to apply to the at least one artifact.”

This limitation of the claim requires that the presence of a data structure (or file) that specifies one of the read consistency and the write consistency to apply to the artifact (*see e.g.*, Specification, Code Samples 1-7). Turning to rejection, the Examiner has asserted that the following portion of Jensen teaches the this limitation (*see* Office Action mailed July 24, 2006, p. 3):

Second, it ensures that only one copy of an object instance is in the cache at any given time, even if several different queries return the same information from the database. Third, the mechanism guarantees the integrity of data in the cache by locking data appropriately in the structured database during a database transaction, flushing cache data at the end of each transaction, and transparently re-reading the data and reacquiring the appropriate locks for an object instance whose data has been flushed. (Jensen, col. 4, ll. 41-49)

A review of the portion of Jensen cited by the Examiner reveals no disclosure of a consistency specification (or any other data structure) that specifies one of the read consistency and

the write consistency to apply to the artifact. In fact, the portion of Jensen cited by the Examiner is directed to ensuring that the integrity of data in the cache without any disclosure of a specification that specifies one of the read consistency and the write consistency to apply to the artifact, where the artifact is used by the application.

Moreover, in response to the Examiner's assertion that, "...locking data...suggests that the data is read only consistent, so the data can be assessed based on whether the data is locked or not, in other words read or write" (*see* Office Action mailed July 24, 2006, p.10), Applicant respectfully asserts that the fact that Jensen supports the functionality to lock data overlooks the fact that the claims are directed to a consistency specification that defines the *threshold* decision of whether or not to lock the data. Said another way, the consistency specification defines how an application should use locking mechanisms to control the access of the data in a database, whereas Jensen is limited to disclosure of the locking mechanism itself.

Jensen fails to disclose using a consistency specification to obtain data when the application enters a particular state

Independent claim 1 additionally requires, in part, "wherein the application accesses data from the database associated with the at least one artifact using a consistency specification when the application enters the at least one of the plurality of the states."

This limitation requires that the application accesses data associated with the artifact in accordance with the consistency specification when the application enters a particular state. Turning to the rejection, as discussed above, Jensen fails to disclose a state, where the state corresponds to a

state within the application. Further, as discussed above, Jensen fails to disclose a consistency specification. In view of Jensen's failure to disclose a state (as recited in the claims) and a consistency specification, it logically follows that Jensen also does not disclose using the consistency specification when entering a particular state within an application.

Conclusion

As Jensen does not disclose each and every element recited in independent claim 1 of the '884 Application, Jensen is not a proper anticipating reference under 35 U.S.C. §102(b). *See Brown*, 265 F.3d at 1351. Accordingly, Jensen is also an improper anticipating reference for claims 2-9 and 12, which depend, either directly or indirectly, from claim 1. Withdrawal of this rejection is respectfully requested.

B. Claims 13-17

Claim 13 is representative of this group. Accordingly, the following arguments with respect to claim 13 are similarly applicable to claims 14-17.

Jensen does not disclose a state or a transition

Independent claim 13 requires, in part, "obtaining an application usage specification that defines the application as a plurality of states and a plurality of transitions, wherein the at least one artifact is associated with one of the plurality of states."

The above limitation clearly requires that an application includes a state and a transition, where an artifact is associated with the state. A state defines an interaction with a client, and as claimed, corresponds to the state of the application (*see, e.g.*, Specification, paragraph [0028]). The Examiner asserts that in part, the following portions of Jensen teach this limitation (*see* Office Action mailed July 24, 2006, page 6):

... and a state of 1 in attribute 213. The reference count of 1 indicates that only one variable in the object-oriented application refers to employee instance 211. Employee instance 218 contains a reference count of 1 and a state of 0 in attribute 220 (Jensen, col. 9, lines 22-31)

However, the cited portion of Jensen refers to attributes and states of an object instance. As discussed above, an object instance corresponds to an instantiation of an object class, which is “a set of data (attributes) and functional capabilities (routines) encapsulated into a single logical entity” (*see* Jensen, col. 5, lines 45-49). The term “state” as used by Jensen corresponds to the state of any given piece of data of an object instance, and relates the data to other data (*see* Jensen, col. 9, lines 20-26). In contrast, “state,” as recited in the claims, corresponds to the state of the application. Thus, the term “state” used in Jensen is clearly not equivalent to the term “state” recited in the claims.

Jensen does not disclose obtaining a consistency specification

Independent claim 13 additionally requires, in part, “obtaining a consistency specification that corresponds to at least one transaction, wherein the at least one transaction comprises at least one of the plurality of states and one of the plurality of transitions and the consistency specification

includes at least one of a read consistency and a write consistency to apply to the at least one artifact.”

The above limitation requires that the consistency specification (*e.g.*, a data structure or file) specifies one of a read consistency and a write consistency to apply to an artifact (*i.e.*, a variable, relationship, attribute, etc.) within a transaction (*see, e.g.*, Specification, paragraph [0027], Code samples 1-7). As discussed above, a consistency specifies how variables are to be read from or written to. The Examiner asserts that the following portions of Jensen teach this limitation (*see* Office Action mailed July 24, 2006, pages 6):

... indicating that the data for these instances must be re-read from the database and the corresponding locks reacquired to ensure consistency of the data in the cache (Jensen, col. 12, lines 13-15)

... The query (or queries) retrieves the appropriate information to update the data in the object instance and reacquires the appropriate database locks. The method then uses the row or rows returned to update the information in the object instance (Step CF) and sets (assigns) the state of the object instance to valid (Step CG), indicating that the data in the object instance is valid, that is to say, guaranteed to be consistent with the corresponding information in the database (Jensen, col. 13, lines 1-9)

It would be clear to one skilled in the art that the portions of Jensen cited by the Examiner reveal no disclosure of a consistency specification, or any data structure or file, which specifies one of the read consistency and the write consistency to apply to the artifact. In contrast, the portions of Jensen cited by the Examiner are directed to ensuring data integrity in a cache using a standard locking mechanism, and updating data in a database and ensuring internal consistency of the data, respectively. Said another way, the portions of Jensen cited by the Examiner merely disclose a mechanism for ensuring data integrity in a cache, but are completely silent with respect to a data

structure (*i.e.*, the consistency specification) that defines how to apply the aforementioned mechanism for a given artifact based on the state of the application.

Jensen does not disclose generating an application

Independent claim 13 additionally requires, in part, “generating the application using a database schema, the application usage specification, and the consistency specification.” The Examiner has asserted that the following portion of Jensen teaches this limitation (*see* Office Action mailed July 24, 2006, page 6):

This mechanism uses an object model, database schema, and transform to define a mapping between the structured database and object instances of the application. Given these three inputs, it is possible to construct an object-oriented application that can retrieve information from the structured database according to the semantics of the object model. In particular, the application can retrieve a single object instance (that is, retrieve database information corresponding to a single object instance) using an object ID value, and can retrieve object instances related to a given object instance by following the relationship semantics of the object model and using foreign key mappings as specified by the transform (Jensen, col. 10, ll. 46-57)

A review of the portion of Jensen cited by the Examiner reveals that there is no disclosure of using a database schema, an application usage specification, and a consistency specification to generate an application. More specifically, neither the object model nor the transformation corresponds to the consistency specification. The claim recites that the consistency specification “includes at least one of a read consistency and a write consistency to apply to the at least one artifact.” In contrast, Jensen defines an object model as follows:

An “object model” is a set of object classes that together form a blueprint for building an object-oriented application. Each object

class of an object model can have attributes, inheritances, and relationships. (Jensen, col. 5, l. 66 - col. 6, l. 2)

From the above, it is clear that an object model does correspond to the consistency specification.

Further, the “transform to define a mapping between the structured database and object instances of the application” (Jensen, col. 10, ll. 47-48) is also not equivalent to the consistency specification, because the consistency specification defines at least one of a read consistency and a write consistency to apply to the artifact. No such teaching is found in the cited prior art.

Jensen does not disclose that the application accesses data from the database ... using a consistency specification

Independent claim 13 further requires, in part, that “the application accesses data from the database associated with the at least one artifact using a consistency specification when the application enters the at least one of the plurality of the states.”

As discussed above, Jensen fails to disclose a consistency specification, as well as a state, as required by the claimed invention. A consistency specifies how variables are to be read from or written to (*see, e.g.,* Specification, paragraphs [0028]-[0029]). Examples of consistency specifications may be found, for example, in Code Samples 1-7 of the Specification.

The Examiner asserts that the following portions of Jensen teach this limitation (*see* Office Action mailed July 24, 2006, page 3):

Second, it ensures that only one copy of an object instance is in the cache at any given time, even if several different queries return the same information from the database. Third, the mechanism guarantees the integrity of data in the cache by locking data appropriately in the structured database during a database transaction, flushing cache data at the end of each transaction, and

transparently re-reading the data and reacquiring the appropriate locks for an object instance whose data has been flushed (Jensen, col. 4, lines 41-49)

In the illustrated embodiment, each object instance also contains a reference count and state. Department instance 201 has attribute 203 which contains a reference count of 2, indicating that two variables in the object-oriented application refer to department instance 201, and a state of 1, indicating that the data associated with department instance 201 is valid, i.e., has been read since the last database transaction was committed. Employee instance 211 contains a reference count of 1 and a state of 1 in attribute 213. The reference count of 1 indicates that only one variable in the object-oriented application refers to employee instance 211. Employee instance 218 contains a reference count of 1 and a state of 0 in attribute 220. The state of 0 indicates that the data associated with employee instance 218 has been flushed, i.e., has not been read since the last database transaction was committed. The reference count and state can be omitted in some embodiments (Jensen, col. 9, lines 23-35)

A review of the aforementioned portions of Jensen reveals that there is no disclosure of using a consistency specification to determine how to obtain data from a database when an *application enters a given state*; rather, the aforementioned portions of Jensen only teach that each object instance may be associated with a state. Moreover, even assuming *arguendo* that the aforementioned portions of Jensen teach the state of an application, there is no indication that the disclosed states are used to determine how to obtain data from a database when the state is entered. In contrast, the states merely serve to track the current status of the object instance (e.g., whether or not the data in the object instance is valid) without any indication that such information is used to dictate how to obtain data associated with the object instance.

Conclusion

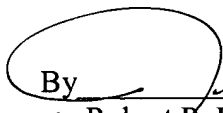
As Jensen does not disclose each and every element recited in independent claim 13 of the '884 Application, Jensen is not a proper anticipating reference under 35 U.S.C. §102(b). *See Brown*, 265 F.3d at 1351. Accordingly, Jensen is also an improper anticipating reference for claims 14 and 15, which depend from independent claim 13. Further, Jensen is an improper anticipating reference for independent claims 16 and 17. Withdrawal of this rejection is respectfully requested.

Conclusion

Applicant believes this reply is fully responsive to all outstanding issues and believes the included amendments simplify issues for appeal, and accordingly, Applicant respectfully requests entry and favorable consideration thereof. If this belief is incorrect, or other issues arise, the Examiner is encouraged to contact the undersigned or his associates at the telephone number listed below. Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference Number 03226/305001).

Dated: December 26, 2006

Respectfully submitted,

By  # 33,986
for Robert P. Lord
Registration No.: 46,479
OSHA · LIANG LLP
1221 McKinney St., Suite 2800
Houston, Texas 77010
(713) 228-8600
(713) 228-8778 (Fax)
Attorney for Applicant